Knowledge-Based Data Management

Reagan W. Moore
San Diego Supercomputer Center

moore@sdsc.edu
http://www.npaci.edu/DICE

Data Intensive Computing Environment

Staff

- Reagan Moore
- Chaitan Baru
- Charles Cowart
- Amarnath Gupta
- George Kremenek
- Bertram Ludäscher
- Richard Marciano
- Arcot Rajasekar
- Michael Wan
- Ilya Zaslavsky
- Bing Zhu

Students

- Pratik Mukhopadhyay
- Azra Mulic
- Kevin Munroe
- Paul Nguyen
- Michail Petropolis
- Nicholas Puz
- Pavel Velikhov
- +/- NN

Topics

- **Requirements driving development of data handling systems - 7 Scenarios**
  - Corresponding features of the SDSC Storage Resource Broker
- **Future management requirements**
  - Knowledge
  - Information
  - Data

Data Handling Infrastructure

- **Model-Based Knowledge Management**
  - Rule-based ontology mapping, conceptual-level mediation - **CMIX**
- **Data Grid**
  - Data federation across multiple libraries - **MIX**
- **Digital Library**
  - Interoperable services for information discovery and presentation - **SDLIP**
- **Data Collection**
  - Information Management - **MCAT**
- **Data Handling**
  - Systems for data retrieval from storage systems - **SRB**
- **Persistent Archives**
  - Storage of data collections for “the life of the republic” - **HPSS**

Scenario # I

- **Data Assimilation Office**
- **Data sets are stored on Unitree at NASA Goddard**
- **Computation is done at NASA Ames**
- **Re-analyze 10 years of data with a new data assimilation code**
- **Can the analysis be automated?**

Collection-based Data Access

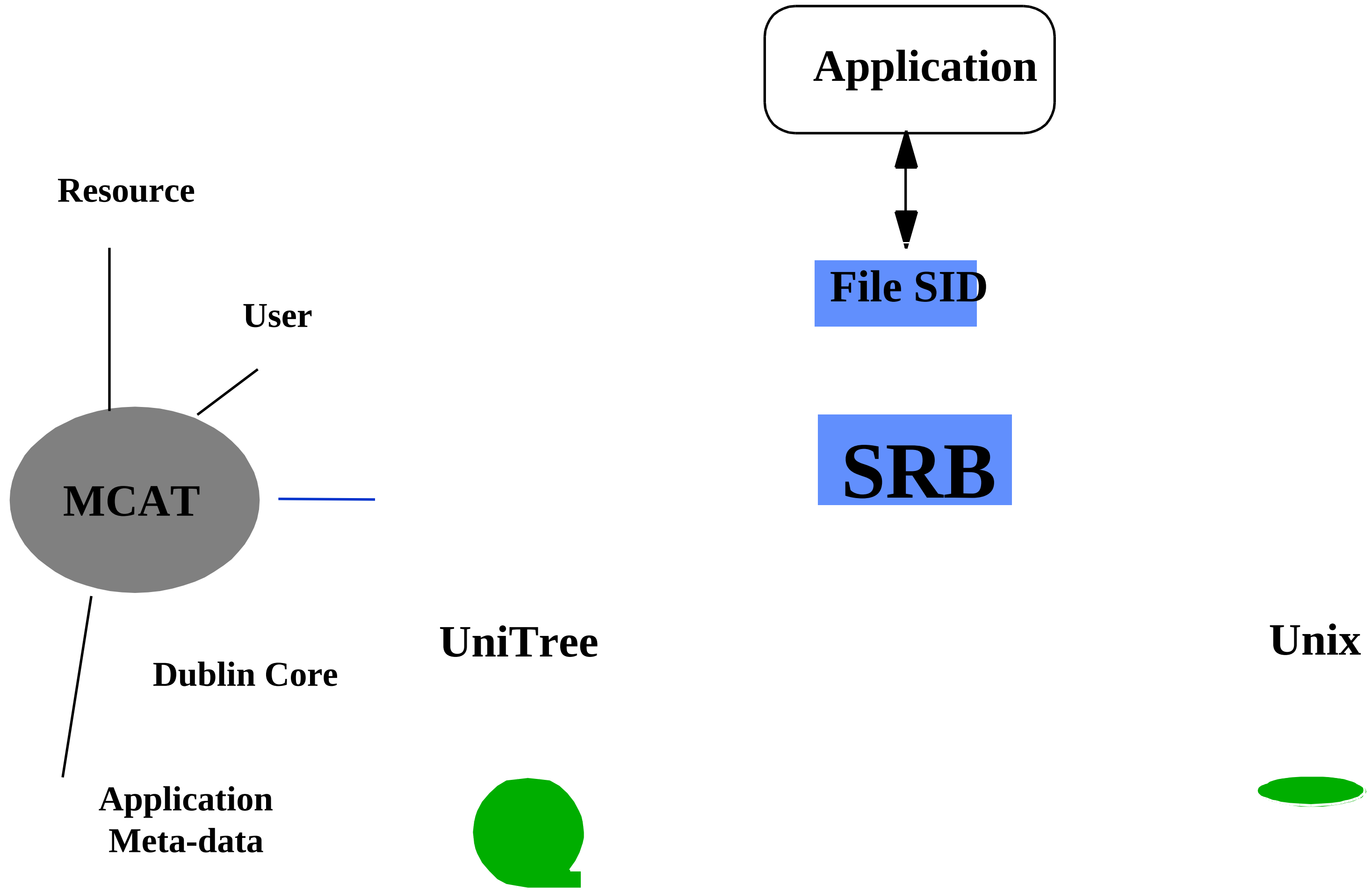
- **Automate access through creation of a collection**
  - Data sets managed by attributes
  - Data set discovery through use of attributes

- **Access across administration domains**
  - Connection to remote resource
    - clConnect call
  - Unix file system semantics
    - srbObjCreate / srbObjClose / srbObjRead / srbObjWrite

Collection Attributes

- **SRB location attributes**
  - Storage location (IP address), Access protocol, local file name
- **Unix file attributes**
  - Owner, creation date, size, access control list, …
- **Dublin core attributes**
  - Provenance information
- **Domain specific attributes**
  - Attributes that describe the digital object

SDSC Storage Resource Broker & Meta-data Catalog - Simplest usage model



An example for creating a high-level data object and writing to it.

The object is created in a storage resource named "unix-sdsc". This resource must have already been registered in MCAT.

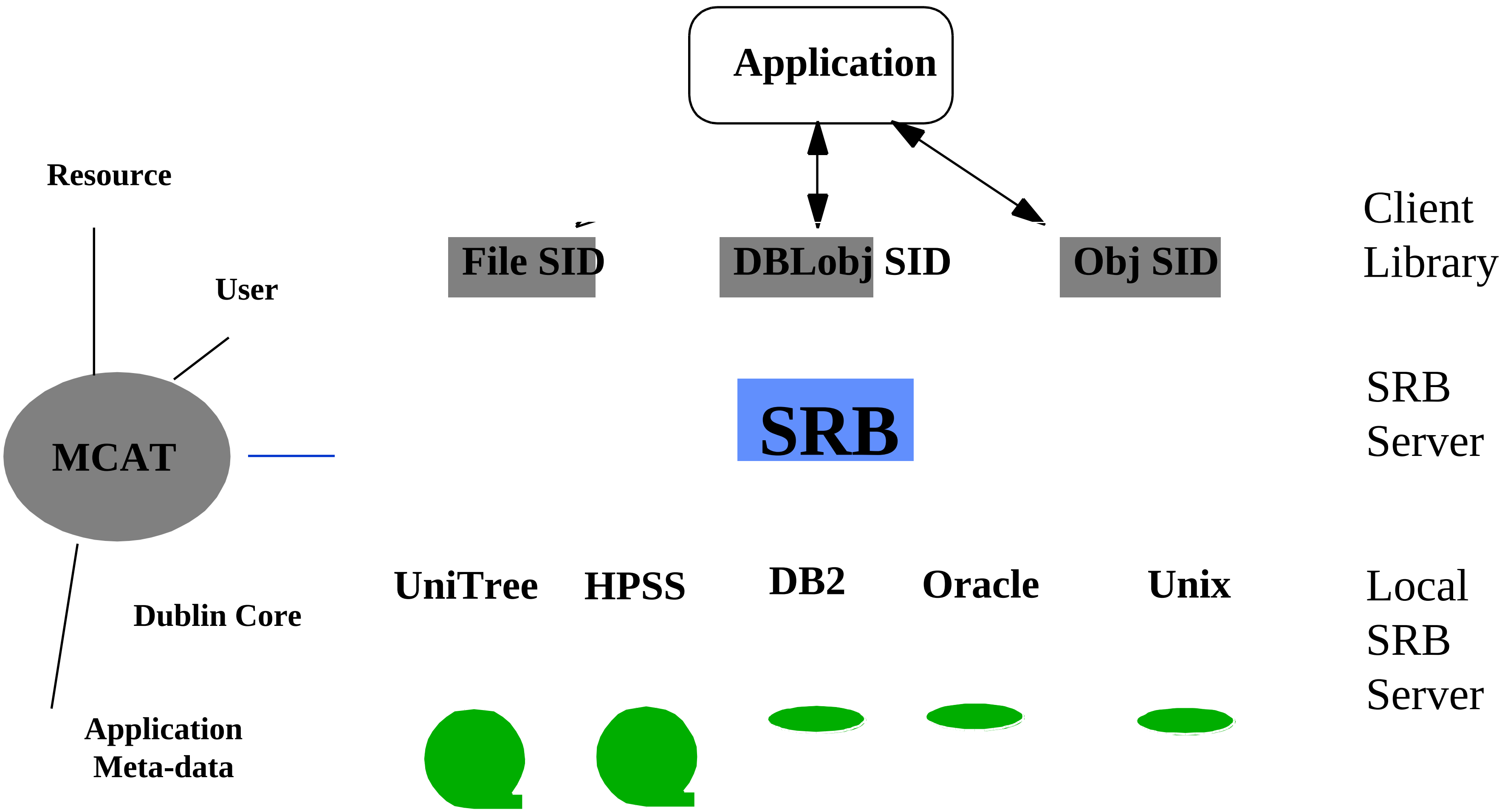
```
#define DATATYPE "ascii text"
#define RESOURCE "unix-sdsc"
#define COLLECTION "/srbtest"
.
int in_fd, out_fd;
int nbytes, tmp;
char buf[BUFSIZE];
srbConn *conn;
.
/* Connect to the SRB server */

conn = clConnect (HOST_ADDR, NULL ,SRB_AUTH);

/* check to see if the connection was successful */

if (clStatus(conn) != CLI_CONNECTION_OK) {

    fprintf(stderr,"Connection to SRB server failed.\n");

    fprintf(stderr,"%s",clErrorMessage(conn));

    exit_nicely(conn);

}
```

```
/* Create a data object with objID = argv[1] */

out_fd = srbObjCreate (conn, MDAS_CATALOG, argv[1],
DATATYPE, RESOURCE ,COLLECTION, NULL, -1);
if (out_fd < 0) { /* error */
    fprintf(stderr, "can't create obj \"%s\", status = %d\n", argv[1], out_fd);
    fprintf(stderr,"%s",clErrorMessage(conn));
    exit_nicely(conn);
}

/* Open a local file with filename = inFileName. */
in_fd = open (inFileName, O_RDONLY, 0);
if (in_fd < 0) { /* error */
    fprintf(stderr, "can't open file\"%s\"\n", inFileName);
    exit_nicely(conn);
}


/* Read from the local file and write to the just created data object */

while ((nbytes = read(in_fd, buf, BUFSIZE)) > 0) {
    /* Write to the data object */
    tmp = srbObjWrite(conn, out_fd, buf, nbytes);
    if (tmp < nbytes) {
        fprintf(stderr, "Error: Read %d bytes, Wrote %d bytes.\n ",
        nbytes, tmp);
        exit_nicely(conn);
    }
}
srbObjClose (conn, out_fd);
close (in_fd);

 /* Disconnect */
```

Scenario # 2

- **NASA Information Power Grid**
- **Demonstrate distributed data analysis using multiple NASA resources while accessing data objects stored at multiple sites:**
  - HPSS at SDSC
  - File systems at Caltech
  - NASA FTP sites
- **Support access to legacy systems**

SDSC Storage Resource Broker & Meta-data Catalog



Scenario # 3

- **California Digital Library**

- **Provide persistent identifiers (site and access protocol independent)**
- **Provide support for copies of the data objects**
- **Provide archival backup for the data collections**
- **Manage persistence across technology evolution**

Logical Resource Naming

- **Create a logical resource name that groups multiple physical resources**
- **Writing to the logical resource name writes to all of the associated physical resources**
  - Completion on write to “k” of “n” resources, k < n
- **Latency management**
  - Access copy stored on lowest latency storage system

Scenario # 4

- **NSF NPACI - Digital Sky Project**
- **Support formation of a 5-million image, 10-TB image collection for 2MASS**
- **Store images in an archive**
- **Sort images into containers based on spatial location rather than temporal order as seen by the telescope**

Containers

- **Containers are used to aggregate data sets**
  - Minimizes number of files seen by archives
  - Improves latency of access for related files
- **Containers have a maximum size**
  - When write into a container, a new container is automatically started when the initial container is full
  - Metadata catalog manages mapping from object to container
- **Containers are cached on disk after retrieval from archive**

An example for creating a Container and putting an SRB object in it.

The container is created in a storage resource named "cont-sdsc". This resource must have already been registered in MCAT.

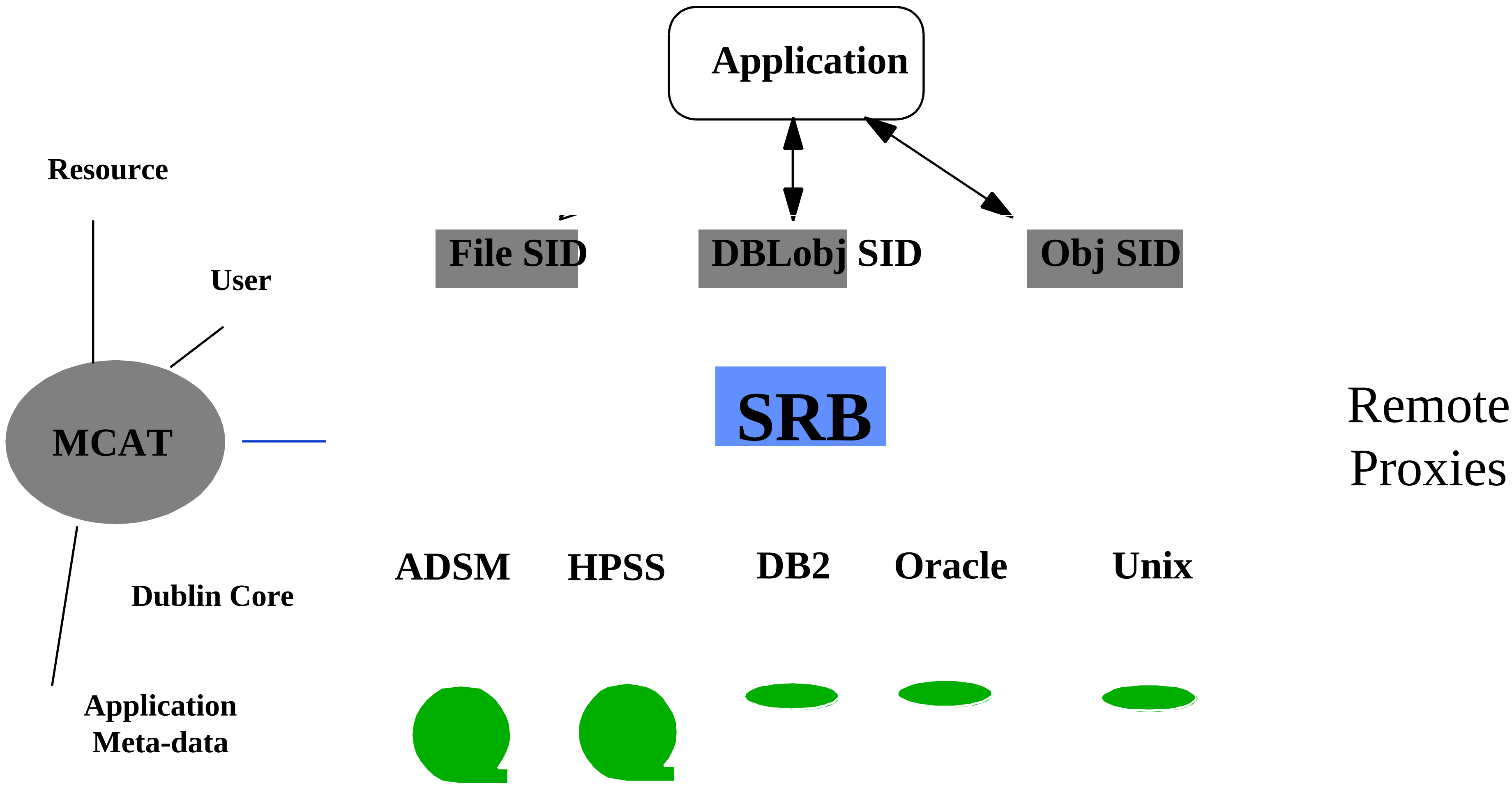
```
#define MDAS_CATALOG 0
#define CONT_SZ 20000000    /* 20 Mb container */
#define CONT_NAME    "testContainer"
#define CONT_RESC "cont-sdsc"
#define COLLECTION "/srbtest"
#define OBJ_NAME "testObj"
.
int in_fd, out_fd;
int nbytes, tmp;
char buf[BUFSIZE];
char tmpName[MAX_TOKEN];
srbConn *conn;
.
/* Connect to the SRB server */

conn = clConnect (HOST_ADDR, NULL , NULL);
```

```
/* Create a container */
status = srbContainerCreate (conn, MDAS_CATALOG, CONT_NAME,
  NULL, CONT_RESC, CONT_SZ);

if (status < 0) {
        fprintf(stderr,"Unable to create container %s, status = %d\n",
          CONT_NAME, status);
        fprintf(stderr,"%s",clErrorMessage(conn));
        clFinish(conn);
        exit (1);
}

/* Create a data object */
sprintf (tmpName,
         "%s&CONTAINER=%-s", OBJ_NAME, CONT_NAME);

out_fd = srbObjCreate (conn, MDAS_CATALOG, tmpName,
NULL, NULL, COLLECTION, NULL, -1);
if (out_fd < 0) { /* error */
    .
}
```

Scenario # 5

- **DOE - ASCI Data Visualization Corridor**
- **Provide interactive visualization of terabyte-sized data objects, retrieved from remote archive**
- **Subset data objects at the storage system**
    - Latency management mechanism
- **Page data as needed from remote source into rendering system**

SDSC Storage Resource Broker & Meta-data Catalog

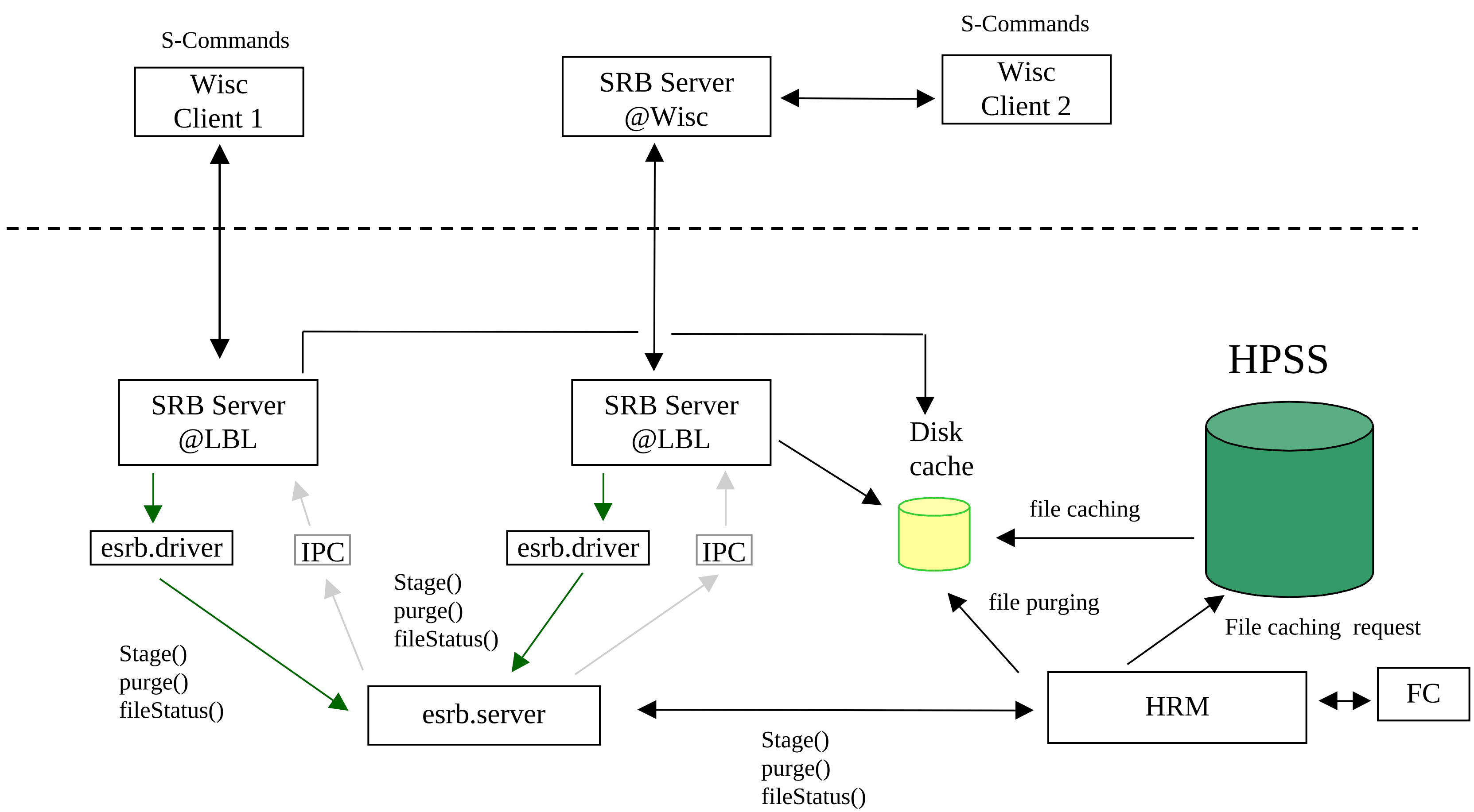


Scenario # 6

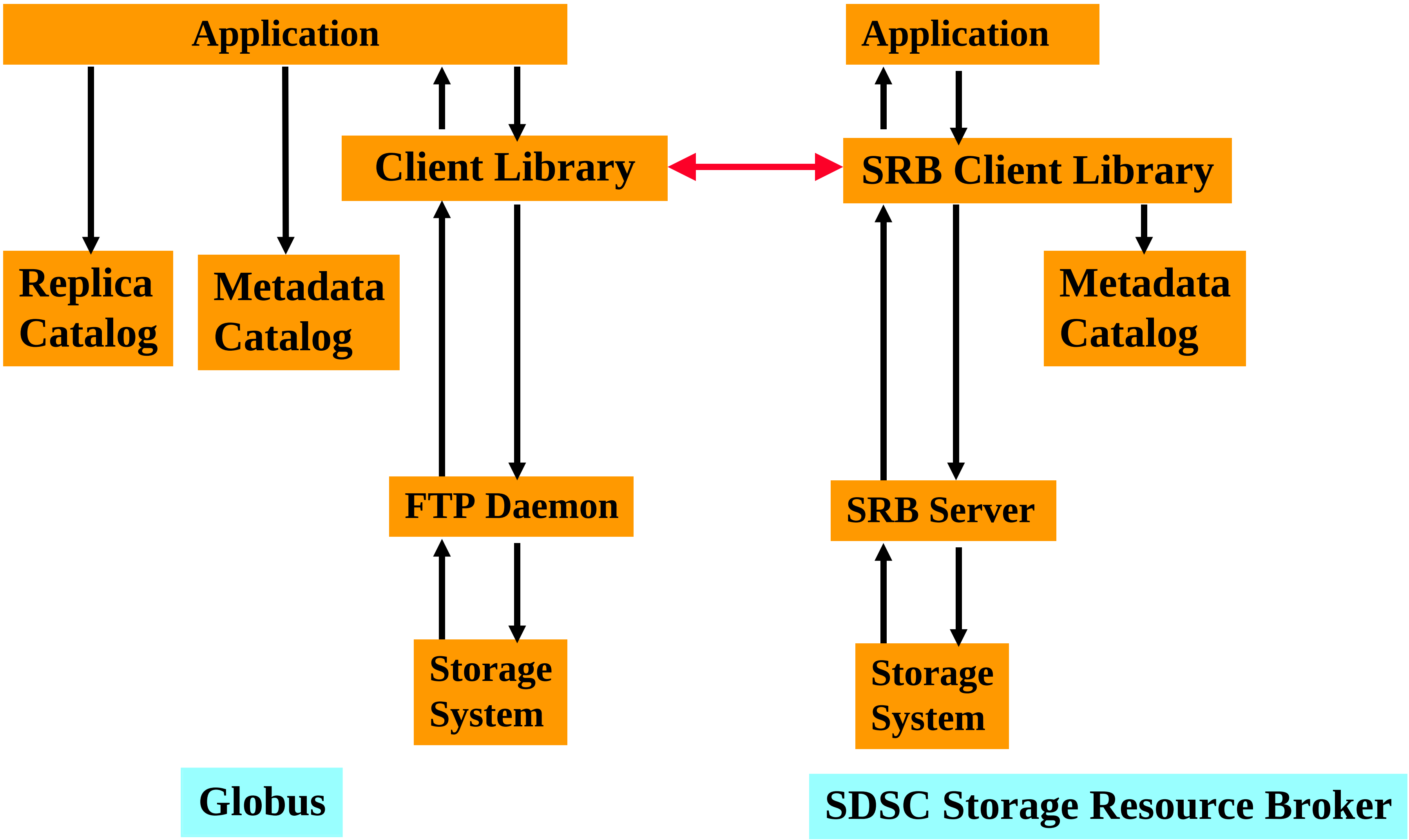
- **DOE - Particle Physics Data Grid**

- **Support replicas of data objects to minimize access latency**
- **Support access to local resource managers**
  - Stage command - to force prefetch
  - Status command - to track progress of resource manager
- **Support interoperation between grids**

Particle Physics Data Grid - Replication System



Interoperation between Grids



Scenario # 7

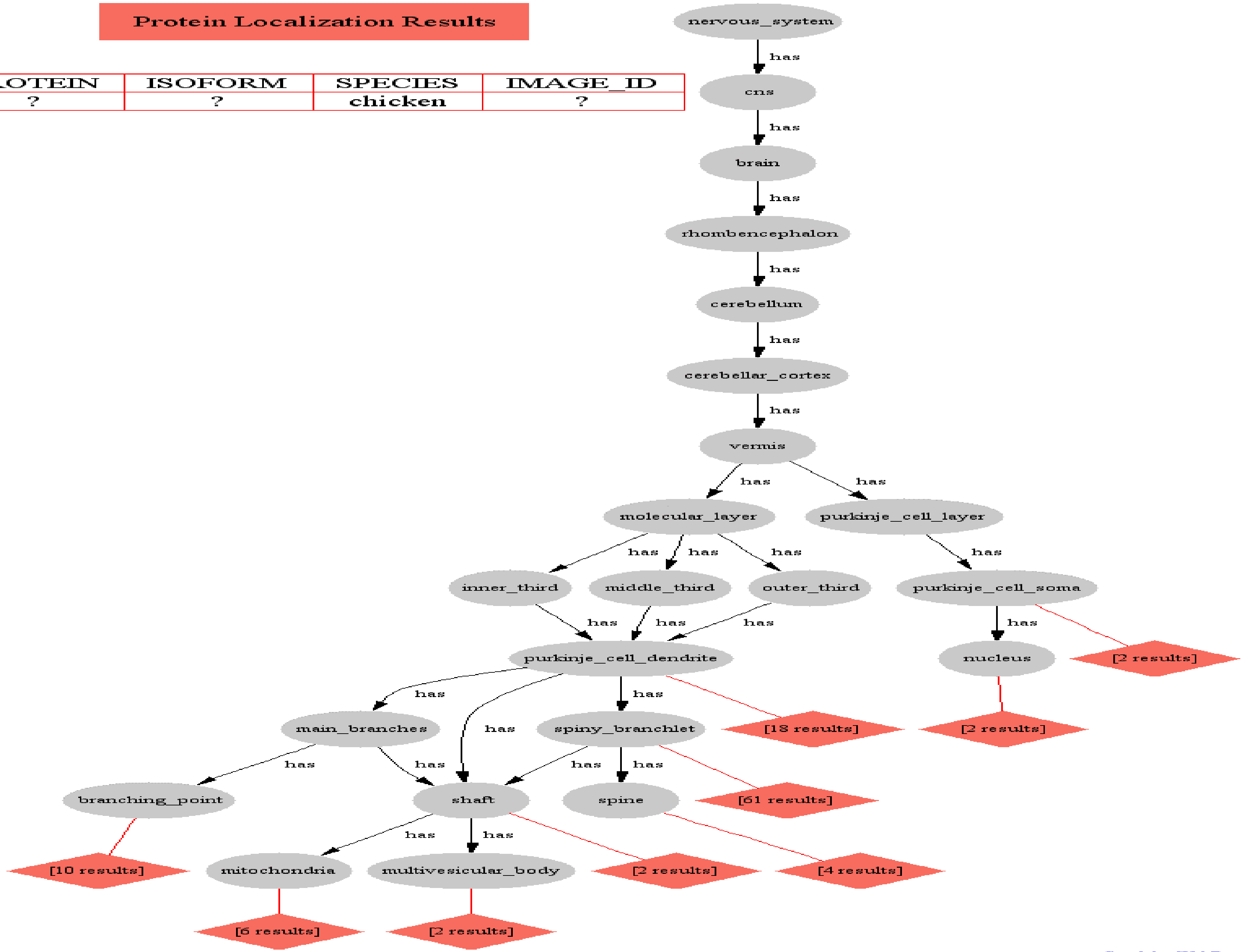
- **NSF NPACI - Neuroscience database federation**
- **Provide rule-based access to multiple data collections**
- **Support access based on domain concepts, rather than collection attributes**
- **Define relationship of retrieved images**

Differentiating between Knowledge, Information, and Data

- **Concepts are tags that describe reality**
  - Examples are nervous systems, neurons, axons, dendrites
- **Metadata are tags that describe attributes of data**
  - Examples are provenance attributes (Dulbin core), or data object parameters
- **Knowledge is the relationship between concepts**
  - Topic Maps (ISO 13250) organize relationships between concepts
- **Information is metadata**
  - Information repositories (databases) organize metadata
- **Data are digital objects that represent reality**
  - Storage systems organize data

Protein Localization Results

PROTEIN	ISOFORM	SPECIES	IMAGE_ID
?	?	chicken	?



Graph by WebDot

Collection Architecture

	Ingest	Manage	Access
Knowledge	Relationships between Concepts	Knowledge Repository for Rules	Knowledge or Topic-Based Query
Information	Attributes Semantics	Information Repository	Attribute- based Query
Data	Fields Containers Folders	Storage (Replicas, Persistent IDs)	Feature-based Query
	Process	Infrastructure	Process

Unified Model for Relationships

- **Relationships quantify rules:**
  - Rules for defining attributes
  - Rules for organizing attributes into schema
  - Rules for feature extraction
  - Rules governing data set creation
- **Relationships quantify associations:**
  - Organization of concepts into topic maps
  - Clustering of data into containers
  - Semantic mapping
  - Ontology mapping

Architecture Interfaces

	Ingest		Manage		Access
Knowledge	Relationships Between Concepts	XTM DTD	Knowledge Repository for Rules	Rules - KQL	Knowledge or Topic-Based Query
	(Topic Maps / Model-based Access)				
Information	Attributes Semantics	XML DTD	Information Repository	EMCAT / MIX	Attribute- based Query
	(Data Handling System - Storage Resource Broker)				
Data	Fields Containers Folders	MCAT/HDF	Storage (Replicas, Persistent IDs)	Grids	Feature-based Query

SRB Functionality Summary #1

Functionality	Benefit to user
SRB ARCHITECTURE	
C/S architecture	Supports remote clients
Distributed SRB servers	Access to distributed storage resources
Distributed storage resources	Access to distributed storage resources
Global namespace	Need to remember only a single, logical name for file/dataset
Support for 64-bit file size	Support up to 2^64 byte files in 64- or 32- bit architectures
Logical storage resources (LSR)	Supports abstraction from physical storage (location xparency), replication, declustering
Capability-based definition of storage resources	Provides abstract definition of storage resource
Replication of data sets	Improved reliability, availability, and performance
Replication across "k" of "n" sites (k<n)	Graceful degradation during network and/or storage system failure
Registration – register "legacy" data set as a SRB object	Ability to manage legacy (pre-existing) data using SRB
Third-party copy operation	Efficient peer-to-peer copying of data
Data partitioning	Partition a data set across multiple storage resources
Containers to support transparent clustering/collocating data sets	Improved response time in accessing multiple, related files; overcomes namespace restrictions imposed by underlying storage resource; efficient utilization of storage resource;
Caching support for containers	Improved response time
Prefetch of files	Provides scheduled, overlapped I/O; improved response time
Support for network of distributed caches	Improved response time
Support for multiple archive resources in containers	Provides redundancy and failover capability

SRB Functionality Summary #2

Functionality	Benefit to user
METADATA CATALOG	
Attributed-based data set identification	Allows search and discovery of data sets via user-defined attributes
Collection-based (hierarchical) management of distributed digital objects	Logical view of data organization is different from physical organization of directory structures
Persistent identifiers for data namespace	Physical independence, location transparency
Support for Dublin core metadata	Use of industry-standard metadata definition
Unix filesystem metadata	Compatibility with UNIX filesystem
User definable metadata for data sets	Provides user-defined attribute-based search of data sets
User defined metadata for collections – 10 text attributes and 2 numeric	Can inherit metadata in collection hierarchy
Support for annotating data sets	Can associate "free form" annotation with data sets/collections
Store and utilize resource metadata, e.g. resource capacity (in bytes), latency, bandwidth	Enables "intelligent" selection of resource and data set
Ability to manage individual users and groups	Provides sophisticated authentication and access control mechanisms
Ability to assign roles to users, e.g. "regular user", Curator, SRB Administrator	Supports multiple roles for individual users, enabling complex, collaboration environments
Customizable attribute sets	Allows a curator to add new attributes to a collection

SRB Functionality Summary #3

Functionality	Benefit to user
CLIENT SUPPORT	
C/C++ client libraries, and utilities, for Sun, AIX, Unicos, Irix, Linux, NT, NT/2000	Can log in from a variety of clients
Client GUI browser for NT/2000	Provides familiar folder abstraction for collections and data sets
Java browser	Provides folder abstraction on Java platforms
Utility to batch-load containers	Allows efficient bulk loading/registration of data
Web interface	Provides Web-based access to metadata and data
SQL interface support for RDBMS storage resources	Allows DBMS-style row-based access to data stored in a database
Seamless broker. Provides "transparency" in accessing data sets.	Enables use of SRB with existing/legacy UNIX utilities and apps. No SRB "learning curve".
Template language support for style sheets	Allows data to be returned as HTML, XML...
Template language rule support for data ingestion into a database	Allows HTML, XML, etc. data to be ingested as rows

SRB Functionality Summary #4

Functionality	Benefit to user
SERVER SUPPORT	
MCAT Server for Sun/Oracle, AIX/DB2, NT/Oracle, DB2v7	Allows use of MCAT
SRB Server for Sun, AIX, Unicos, Irix, NT, Linux	Allows use of SRB
Data resource servers for Unix filesystem, NTFS, DPSS, ADSM, HPSS, UniTree, FTP, HTTP, DB2 UDB, Oracle	Access to heterogeneous storage resources
Resource access control	Provides control over level of access to each resource based on user, group.
Access control over collections and data sets	Provides control over type of access to collections and/or data sets based on user and/or groups
Ticket based access	Provides flexible access control policies
Audit trails for auditing data sets accesses	Provides details on system activity at data set level
Encrypted password-based authentication	Provides support for standard UNIX-style authentication
GSI authentication interoperability	Can authenticate using GSI system
PKI authentication	Can authenticate using PKI
SRB Servelet support (remote proxy operations)	Improves performance due to server-side processing, or "function-shipping"
Metadata for defining SRB Servelet parameters for remote proxies	Allows easy selection and use of SRB Servelets
GUI for system administration	GUI-based admin
SRB monitor for checking and restarting distributed servers	Allows easy monitoring of overall distributed system
Pre-spawned server to improve performance	Improves performance by reducing latency on initial connect to SRB

Further Information

http://www.npaci.edu/DICE